09/446425
420 Rec'd PCT/PTO 20 DEC 1999

1. TITLE

Method and Computer System for Encoding a digital message, for transmission of the message from a first computer unit to a second computer unit, and for decoding the message

2. Technological Background

Various network protocols are known in the area of managing computer networks. The jobs for the management of computer networks are becoming increasingly more difficult due to the great spread of computers and the more and more complex networking of computers and the systems for network management required for this purpose are becoming more and more powerful. The question of security of the network management is acquiring greater and greater significance in the framework of management of computer networks. The security of the network management is highly dependent on the security techniques employed in the system.

The document (M. Rose, The Simple Book, PTR Prentice Hall, 2nd Edition, ISBN 0-13-177254-6, pages 59-91, 1994) discloses various network protocols for network management, for example the Simple Network Management Protocol (SNMP) in Version 1 (SNMPv1) and in Version 2 (SNMPv2) or the Common Management Internet Protocol (CMIP) as well.

The SNMPv1 has been hitherto most widespread for monitoring and supervision of network components both over local computer networks (Local Area Networks, LANs) as well as given global networks (Wide Area Networks, WANs).

The SNMPv1 is arranged above the Internet protocols of user datagram protocol (UDP) and Internet protocol (IP) in the framework of the OSI Communication Layer system. Both the UDP as well as the IP exhibits substantial weaknesses in the area of security, since security mechanisms are hardly integrated through not at all integrated in these protocols.

Below, both the SNMP as well as CMIP are referred to as network protocol.

The network protocols are employed for the transmission of computer network management information between a first computer unit, which contains what is referred to as a manager, and at least one second computer unit, which contains what is referred to as an agent. In a complex computer network, at least one management station and an arbitrary plurality of computers monitored and supervised by the manager application are usually monitored or, respectively, controlled via the network protocol.

However, network management architectures are also known that comprise a plurality of hierarchies, for example a plurality of computers that are respectively monitored by one manager, and a plurality of computers that respectively contain a manager application that are monitored or, respectively, controlled in turn by a further computer that contains a higher-ranking manager application.
A computer that contains a manager application of the respective network protocol is referred to below as first computer.

Each computer unit that has an agent implemented is referred to below as second computer unit.

It is possible that a computer is configured both as manager as well as as agent; correspondingly, the functionalities are contained in the computer.

The respective network protocol can be realized in the computer both in hardware as well as in software.

A simple hierarchy is assumed below, i.e. only that case is described wherein a first computer as manager monitors or, respectively, controls an arbitrary plurality of second computers, the agents. This, however, only serves for the purpose of a simpler presentation. It is possible without further ado to also apply the invention in an architecture having an arbitrary plurality of hierarchy levels.

In the network protocols, either an information query is transmitted from the first computer unit to the second computer units or a control value is transmitted for the control or, respectively, supervision of the second computer unit.

It is standard in each second computer unit given the known network protocols that the information employed by the second computer unit in the framework of the network protocol is stored in the form of what is referred to as a management information base (MIB), which exhibits the structure of a hierarchic data bank.

The overall structure of the management information of the network protocols is stored in what is referred to as a global registration tree, for example the global SNMP registration tree. The MIB of an agent, i.e. of a second computer unit, is a part of the registration tree of the respective network protocol.

Digital messages, for example an SNMPv1 message, are employed for the transmission of information between the first computer unit and the second computer unit.

An SNMPv1 message contains a version number, what is referred to as a community string, and an SNMPv1 protocol data unit (PDU).

The version of the network protocol employed is indicated with the version number. The version number is defined upon implementation of the respective network protocol.

The community string in the SNMPv1 serves as password for access to an MIB of a second computer unit. The community string given SNMPv1 is sent to the agent unencrypted. A check is carried out in the agent, i.e. the second computer unit, to see whether the community string that was respectively received together with an SNMPv1 message authorizes an access in the MIB of the second computer. Since the password is transmitted unencrypted given SNMPv1, a misuse of the community string is easily possible, for example for masking a potential attacker and for unauthorized access to a second computer unit, since it is very simple for a potential attacker to tap the community string together with an IP sender address of an authorized user.

SNMPv1 thus has practically no effective security mechanism integrated in it, particularly no effective authentification of the SNMPv1 manager, and, as a

consequence of the lacking authentification, no dependable access control on the part of the agent. Further, SNMPv1 contains no possibility for implementing security mechanisms of the data integrity or of the data confidentiality. It is thus possible without further ado for a potential attacker to simply listen in to transmitted SNMP-PDUs and to misuse the transmitted information between manager and agent.

The encoding rules of the network protocols are described in detail in M. Rose, The Simple Book, PTR Prentice Hall, 2nd Edition, ISBM 0-13-177254-6, pages 59-91, 1994.

In the second version of SNMP, SNMPv2, various security measures were in fact provided but, in particular, the administration of cryptographic keys was so involved that this problem led to the fact that the SNMPv2 was incapable of prevailing in the marketplace over the SNMPv1 despite considerably greater possibilities for the administration of computer networks compared to SNMPv1. The original SNMPv2 standard was therefore withdrawn and replaced by a modified standard wherein no security was integrated.

CMIP, which due to generally significantly greater complexity compared to SNMPv1 and SNMPv2, was hardly considered in products was incapable of prevailing in the marketplace.

Further, the concept of what is referred to as proxy agents is likewise described in the document of (M. Rose, The Simple Book, PTR Prentice Hall, 2nd Edition, ISBN 0-13-177254-6, page 315, 1994).

3. Brief Description of the Invention

The invention is thus based on the problem of specifying methods as well as a computer system for encoding, transmission and decoding of a digital message, whereby cryptographic security mechanisms are provided that are simpler than in the known methods and arrangements.

Given the method according to patent claim 1, a digital message that is to be transmitted from the first computer unit to the second computer unit is encoded into an encoded message upon employment of an encoding format of a network

protocol. The encoded message is subjected to at least one cryptographic process and the cryptographically processed, encoded message is again encoded upon employment of the encoding format of the network protocol.

Given the message according to patent claim 2, the message is decoded according to the encoding format of the network protocol. Further, the decoded, cryptographically processed message is subjected to a cryptographic method inverse relative to the at least one cryptographic method, and the inversely cryptographically processed message is decoded according to the encoding format of the network protocol.

Given the method according to patent claim 3, a digital message that is to be transmitted from the first computer unit to the second computer unit is encoded into an encoded message upon employment of an encoding format of a network protocol. The encoded message is subjected to at least one cryptographic process and the cryptographically processed, encoded message is again encoded upon employment of the encoding format of the network protocol. After the encoding has ensued, the entire message is transmitted from the first computer unit to at least the second computer unit. The received message is decoded in the second computer unit according to the encoding format of the network protocol. Subsequently, the decoded message is subjected to the cryptographic process inverse relative to the cryptographic process employed. In a last step, the inversely cryptographically processed message is decoded according to the encoding format of the network protocol.

As a result of the "double" encoding or, respectively, decoding with the respective network protocol, a very simple solution conforming to the standards is proposed in order to cryptographically secure the transmission of messages of a network protocol.

The method also exhibits the considerable advantage of simple realizability and, thus, of fast implementability with the assistance of a computer. A further advantage may be seen therein that the network protocols can remain unmodified and no new network protocols need be defined. Thus, no complicated

version switching or even redefinition of network protocols is required. The cryptographic security of the respective network protocol can be substantially enhanced without greater outlay.

The computer system according to patent claim 12 contains at least one computer unit that is configured such that the method according to one of the claims 1 through 11 is implemented.

The computer system according to patent claim 13 for encoding a digital message upon employment of an encoding format of a network protocol comprises at least the following components:

- a first means for encoding the digital message upon employment of the encoding format of the network protocol to form an encoded message;
- a second means for the cryptographic processing of the encoded message;
- a third means for encoding the cryptographically processed message upon employment of the encoding format of the network protocol.

The computer system according to patent claim 14 for decoding a digital message that is present in an encoding format of the network protocol comprises at least the following components:

-- a fifth means for receiving the encoded, cryptographically processed message from the first computer unit;
-- a sixth means for decoding the received message according to the encoding format of the network protocol;
-- a seventh means for the inverse cryptographic processing of the decoded, cryptographically processed message; and
-- an eighth means for decoding the inversely cryptographically processed message according to the encoding format of the network protocol.

The computer system according to patent claim 15 for encoding a digital message, for transmitting the message from a first computer unit to a second computer unit and for decoding the message contains at least the following components:

- a first computer unit that comprises at least the following components:

-- a first means for encoding the digital message upon employment of an encoding format of a network protocol to form an encoded message;

-- a second means for the cryptographic processing of the encoded message;

-- a third means for the encoding of the cryptographically processed message upon employment of the encoding format of the network protocol;

-- a fourth means for sending the encoded, cryptographically processed message from the first computer unit to the second computer unit;

- a second computer unit that comprises at least the following components:

-- a fifth means for receiving the encoded cryptographically processed message from the first computer unit;

-- a sixth means for decoding the received message according to the encoding format of the network protocol;

-- a seventh means for the inverse cryptographic processing of the decoded, cryptographically processed message; and

-- an eighth means for decoding the inversely cryptographically processed message according to the encoding format of the network protocol.

The computer systems thus likewise exhibit the advantages described above in conjunction with the method.

Advantageous developments of the invention derive from the dependent claims.

The method can be especially advantageously employed in conjunction with SNMPv1 as network protocol, since practically no cryptographic security was previously present for SNMPv1.

However, this method and the corresponding arrangement for the implementation of the method can also be employed in the other network protocols, since the overall complexity of the respective network protocol therein is also considerably reduced.

Further, it is advantageous in the computer system to fashion a second means for cryptographic processing of the encoded message, a third means for

encoding the cryptographically processed message upon employment of the encoding format of the network protocol as well as a fourth means for sending the encoded, cryptographically processed message to the second computer unit as what is referred to as a proxy agent, which is connected to the first means for encoding the digital message upon employment of the network protocol via a communication connection that is assumed to be secure. The first proxy agent and the first computer unit can be realized in common in one computer unit or can also be realized in two different computer units.

In this way, the realization of a computer system for cryptographically secure transmission of messages of the encoding format of a network protocol is achieved upon employment of the proxy technique, which is known from the document of (M. Rose, The Simple Book, PTR Prentice Hall, 2nd Edition, ISBN 0-13-177254-6, page 315, 1994).

This advantage can likewise be established when a fifth means for the reception of the encoded, cryptographically processed message, a sixth means for the decoding of the received message according to the encoding format of the network protocol as well as a seventh means for the inverse cryptographic processing of the decoded cryptographically processed message are realized together in a second proxy agent that is connected to the agent of the second computer unit upon employment of the network protocol via a communication connection assumed to be secure.

4. Brief Description of the Figures

The figures show an exemplary embodiment of the invention, which is explained in greater detail below.

Shown are:

Figure 1 a flowchart wherein the inventive method is shown with realization details for a get request;

Figure 2 a flowchart wherein the method is shown in terms of its method steps with realization details for a set request;

Figure 3 a flowchart wherein the method is shown in abstract form;

Figure 4 a schematic illustration of a possible structure of a cryptographically processed SNMPv1 message wherein the security mechanism of authentification of the original data is realized;

Figure 5 the structure of a possible, cryptographically processed SNMPv1 message with which the security services of integrity and confidentiality of the transmitted SNMPv1 message is realized;

Figure 6 the possible structure of a cryptographically processed SNMPv1 message wherein the security service of confidentiality of the SNMPv1 message is realized.

5. Figure Description

Get-Request

Figure 1 symbolically shows a first computer unit C1 and a second computer unit C2. The first computer unit C1 comprises a manager application MA of the SNMPv1 as well as a first proxy agent PA1.

The second computer unit C2 comprises an SNMPv1 agent AG as well as a second proxy agent PA2 at the side of the second compute C2.

In a first step 101, a get request is formed in the first computer unit C1. What is to be understood by formation of a get request is that a digital message is encoded upon employment of an encoding format of the SNMPv1 network protocol to form an encoded message, the get request. This ensues in a first means 101 of the first computer unit C1 for encoding the digital message upon employment of the encoding format of the network protocol.

In a second step 102, the get request, i.e. the encoded message CN, is sent from the first means M1 to the first proxy agent P1 at the side of the first computer unit C1.

In a third step 103, the encoded message CN is received in the first proxy agent PA1.

In a fourth step 104, the encoded message CN is subjected to at least one cryptographic process in the first proxy agent PA1. A second means 104 is utilized for the cryptographic processing of the encoded message in the fourth step 104.

What is to be understood by a cryptographic method is any arbitrary cryptographic method, for example for authentification, for securing the data integrity or for encryption of digital data as well. For example, the RSA method or the data encryption standard as well, which is referred to as DES method, can thereby be employed.

As a result, one obtains a cryptographically processed message KBN whose format is shown, for example, in Figures 3 through 6 and explained in greater detail below.

In a fifth step 105, the cryptographically processed message KBN is again encoded upon employment of the encoding format of the SNMP network protocol. What is to be understood by this method step is that the cryptographically processed get request is preferably encoded in a set request, i.e. encapsulated. Further, a third means 105 for the encoding of the cryptographically processed message upon employment of the encoding format of the network protocol is provided.

As becomes clear below, it is advantageous to encode any type of message that is to be transmitted from the first computer unit C1 to the second computer unit C2 as set request in the fifth step 105. This is advantageous since the syntax of SNMPv1 only allows object identifiers as payload data to be transmitted for a get request. It is not possible in SNMPv1 to involve the cryptographically processed information in an SNMP get request.

In a sixth step 106, the set request is transmitted as encoded, cryptographically processed message CKN from the first computer unit C1 to the second computer unit C2, i.e. from the first proxy agent PA1 to a second proxy agent PA2.

The encoded, cryptographically processed message CKN is received in a seventh step 107 by the second proxy agent PA2 of the second computer unit C2. To

this end, a fifth means 107 is provided for the reception of the encoded, cryptographically processed message CKN.

In an eighth step 108, a get response - in conformity with standards - is sent from the second proxy agent PA2 to the first proxy agent PA1 of the first computer unit C1 as reply to the set request. The get response contains the respective error status as confirmation.

In a ninth step 109, the received, encoded, cryptographically processed message CKN is de-encapsulated, i.e. decoded, upon employment of the encoding format of the network protocol. A sixth means 109 is provided for the decoding of the received message corresponding to the encoding format of the SNMPv1 protocol.

In a tenth step 110, the second proxy agent PA2 applies the cryptographic process inverse relative to the respectively provided cryptographic process, for example for authentification, for decryption or, respectively, for securing the integrity of the transmitted data, onto the decoded, cryptographically processed message DKN. A seventh means 110 for the inverse cryptographic processing of the decoded, cryptographically processed message DKN is provided for this purpose.

Further, the inversely cryptographically processed message IKN, i.e. the original get request, is sent from the second proxy agent PA2 to the agent application AG of the second computer unit C2.

In an eleventh step 111, the get request is received by the agent AG. An eighth means 111 for reception of the get request is provided for this purpose.

In a further step 112, the inversely cryptographically processed message is decoded according to the encoding format of the SNMPv1 protocol to form the digital message, i.e. is interpreted. This means that, for the specific instant of the get request, the information requested via the get request, namely of a value of what is referred to as a managed object (MO) that is stored in the MIB of the agent AG, is read out. The particular as to what information is in fact requested is contained in the original get request as object identifier.

The requested action, the read out of the requested information in this case, a value of a managed object, is thus implemented in the twelfth step 112. To this end, a ninth means 112 is provided for the implementation of the requested action.

As provided in SNMPv1, a get response is formed by the agent AG in the second computer unit as reply to a get request and, in a thirteenth step 113, is sent to the second proxy agent PA2. The get response contains the result of the action that was requested by the first computer unit C1 in the get request.

The get response is referred to below as reply message AN. The reply message AN can be transmitted either directly to the first computer unit C1 or, for further enhancement of the cryptographic security, can be encoded again in conformity with the encoding format of the network protocol. A tenth means 112 for sending the result of the action to the first computer unit C1 is provided in the second computer unit C2.

Further, an eleventh means 113 is provided for forming the reply message AN that contains the result of the action and for encoding the reply message AN according to the encoding format of the SNMPv1 protocol.

In a fourteenth method step 114, the second proxy agent PA2 receives the reply message AN. A twelfth means 114 for the reception of the reply message AN is provided for this purpose.

In a fifteenth step 115, the encoded reply message AN is subjected to at least one cryptographic process. For this purpose, a thirteenth means 115 is provided for processing the reply message AN with at least one cryptographic process. The result of this method step is a get response encapsulated in a security frame.

The cryptographically processed reply message KBAN is stored in a security MIB in the second processing agent PA2 (step 116). The structure of the security MIB is described in greater detail later.

In order to obtain to the cryptographically processed reply message KBAN, the first proxy agent PA1 of the first computer unit C1 forms a get request,

i.e. a fetch message ABN. To this end, a fourteenth means 117 is provided for forming and encoding the fetch message ABN according to the encoding format of the SNMPv1 protocol, the cryptographically processed reply message KBAN being requested therewith from the second computer unit C2. Further, the encoded fetch message ABN is sent from the first computer unit C1 to the second computer unit C2.

In an eighteenth step 118, the fetch message ABN, i.e. the get request in this case, is received in the second proxy agent PA2 and, in conformity with the standard, the standard get response, which contains the cryptographically processed reply message KBAN in this case, is sent to the first proxy agent PA1. To this end, a fifteenth means 118 is provided in the second computer unit C2 for receiving the fetch message ABN and for encoding the cryptographically processed reply message KBAN requested in the fetch message ABN according to the encoding format of the SNMPv1 protocol, i.e. for encoding the requested get response.

The encoded, cryptographically processed reply message is transmitted from the second proxy agent PA2 to the first proxy agent PA1.

In a further step 119, the encoded, cryptographically processed reply message contained in the standard-conforming get response is received in the first proxy agent PA1. A sixteenth means 119 for receiving the get response is provided for this purpose in the first computer unit C1.

In a further step 120, the get request is decoded, i.e. de-encapsulated, and the get response originally formed by the agent AG of the second computer unit C2 is sent to the manager application MA of the first computer unit C1. A seventeenth means 120 for decoding the get response and for sending the original get response contained in the get response [sic] that contains the requested information is provided for the manager application MA.

In a last step 121, the get response is received by the manager application MA and the requested value is interpreted and stored. An eighteenth means 121 for receiving and evaluating management information is provided for this purpose in the manager application MA.

What is achieved in this way is that a cryptographic securing of the communication becomes possible without great added outlay and without having to modify the method of the SNMPv1 protocol.

Get-Net-Request

For a get next request, which is likewise provided within the framework of the SNMPv1 protocol, the method is implemented in the same way as described for the get request, merely with a modified, correspondingly adapted object identifier for the requested value of the respective managed object.

Set-Request

Figure 2 shows the method for a set request as encoded, digital message CN. For simpler explanation, only the method is described below; the means are correspondingly fashioned such that the individual method steps can be implemented with the computer units C1, C2.

In a first step 201, the set request, i.e. the digital message, is encoded.

In a second step 202, the manager MA of the first computer unit sends the set request, i.e. the encoded message CN, to the first proxy agent PA1.

In a third step 203, the encoded message CN is received by the first proxy agent PA1.

In a fourth step 204, a cryptographic process is applied to the encoded message CN. The result of the cryptographic processing is a cryptographically processed message KBN.

In a fifth step 205, the cryptographically processed message KBN is again encoded upon employment of the encoding format of the SNMPv1 protocol to form an encoded, cryptographically processed message CKN. A set request is again employed for this purpose.

The set request is sent from the first proxy agent PA1 to the second proxy agent PA2 (step 206).

In a seventh step 207, the second proxy agent PA2 receives the set request.

As a reaction to the reception of the set request, the second proxy agent PA2 sends a get response in conformity with the standard that contains the error status as confirmation (step 208).

In a further step 209, the encoded, cryptographically processed message is decoded, i.e. "unpacked". The result is the decoded, cryptographically processed message DKN.

In a tenth step 210, the cryptographic method respectively inverse relative to the cryptographic method employed is applied to the cryptographically processed message DKN. Further, the inversely cryptographically processed message IKN, i.e. the original set request, is sent from the second proxy agent PA2 to the agent AG of the second computer unit C2.

In an eleventh step 211, the agent AG receives decoded, cryptographically processed message and, in a further step 212, the action indicated in the set request is implemented.

As reaction, the agent AG of the second computer unit CU sends the reply message AN in the form of a get response to the second proxy agent PA2 in conformity with the standard (step 213).

In a fourteenth step 214, the second proxy agent PA2 receives the reply message AN.

In a fifteenth step 215, at least one prescribable cryptographic method is again applied to the reply message AN.

The further method steps 216, 217, 218, 219, 220 as well as 221 correspond to the method steps 116, 117, 118, 119, 120 as well as to 121 described in conjunction with a get request method.

The security MIB contains entries that employ the usual syntax for describing managed objects in their structure. Entries in the security MIB are assigned unambiguous object identifiers that are employed for the unambiguous identification of the entries in the security MIB. The object identifiers are registered in the global SNMP-MIB. What is thus achieved is that the purpose and the syntax of

the respective managed object is known. The various entries of the security MIB can, for example, contain either digitally signed, integrity-protected or encrypted management information. Of course, arbitrary combinations of the above-described mechanisms can be entered in the security MIB and, thus, can be taken into consideration in the framework of the method.

A possible exemplary syntax in AS1.1 (abstract syntax notation one) of such a security MIB is presented below.

The syntax of a secure, encapsulated managed object is OCTET STRING. The structure of such an encapsulated managed object is as follows:

```
SecureMO ::=
        SEQUENCE {
                PlainHeader,
                EncapsulatedData
        }
PlainHeader ::=
        SEQUENCE {
                SecurityAssociationID,
                UsedAlgorithms,
                AlgorithmParameters
        }
EncapsulatedData ::= OCTET STRING
        -- signed, encrypted, or integrity protected
        -- ASN.1-encoded data
SecurityAssociationID ::= OBJECT IDENTIFIER
UsedAlgorithms ::= INTEGER (0..7)
        -- value 0 stands for "no security"
        -- value 1 stands for "signed"
        -- value 2 stands for "integrity protected"
        -- value 3 stands for "signed" and "integrity protected"
```

-- value 4 stands for "encrypted"

-- value 5 stands for "signed" and "encrypted"

-- value 6 stands for "integrity protected" and "encrypted"

-- value 7 stands for "signed", "integrity protected" and "encrypted"

AlgorithmParameters ::=

-- necessary parameters for the particular

-- algorithms in use

The value of the parameter UsedAlgorithms is formed according to the following strategy. It can be represented as bit string having the length of 3 bits, whereby the bit of least significance indicates the employment of digital signature ("signed"); the bit having the second lowest significance indicates, for example, whether mechanisms for the protection of the data integrity are provided ("integrity protected"), and the bit having the highest significance describes whether the data were encrypted.

The result of every cryptographic processing of a message can thus be described as bit string having the length 3. The cryptographically processed message is encoded as OCTET STRING. When it is composed of a plurality of bits not divisible by 8, then, however, it can be expanded into an OCTET STRING by employing what is referred to as padding, i.e. filling bits in without semantic significance.

This situation is shown by way of example in a flowchart in Figure 3.

An SNMPv1 request SR is encoded 301 into ASN.1 (encoding rules, syntax definition, ER) according to the rules for encoding of the respective network protocol. The encoded SNMP request CSR, i.e. the encoded message CN, is subjected to the respective cryptographic process in a second step 302. For example, cryptographic keys, parameters for indicating the algorithm employed, as well as additional information, general cryptographic information VI, for the implementation of the respective cryptographic method are thereby employed.

The deriving bit string BS is converted into an OCTET STRING OS by, for example, filling with filler bits in a step 303, for example upon employment of padding PA.

The abstract procedure for the inverse cryptographic processing is correspondingly inversely implemented.

It is advantageous to apply existing functions for the protection of the communication in the framework of SNMPv1 where it is possible and to strengthen these security functions with further cryptographic processes as necessary.

Thus, it is advantageous to employ the concept of community strings in SNMPv1 in the framework of this method as well. In the framework of the concept of a community, groups are defined and access rights for the respective members of the group are allocated to the individual groups. A community and the access rights allocated to the community are part of a configuration of an SNMPv1 agent. It is advantageous to respectively associate communities with specific security mechanisms. Thus, for example, it is possible to assign different cryptographic algorithms, cryptographic keys and corresponding parameters that are respectively employed in the framework of the cryptographic method to members of the community in a community.

Standard-conforming object identifiers are preferably employed as particulars that are to be employed in the cryptographic processes.

In the security configuration, object identifiers are preferably applied to stored cryptographic keys instead of cryptographic keys, these being referred to below as key identifiers. The respective key material is protected better as a result of this procedure.

Further, the respective key material can thereby be more highly protected in that, for example, the data files wherein the cryptographic keys are maintained are encrypted or specific hardware units are provided for the protection of the cryptographic keys, for example chip cards.

The realization details to be respectively employed derive from the security policy, which can vary greatly in conformity with the application.

Authentification of the Data Source

In order to achieve the security service of authentification of the source data, the following information can, for example, be provided in the cryptographically processed message (see Figure 4).

The SNMPv1 request, i.e. the encoded message CN, is encapsulated with the following header or, respectively, trailer information by the cryptographic processing, as a result whereof the cryptographically processed message KBN arises.

An authentification header AH contains a key identifier KID with which the cryptographic key to be respectively employed is indicated via an object identifier, an algorithm identifier AID with which the respective cryptographic algorithm to be applied for authentification is indicated, algorithm parameters AP with which the parameters that are to be employed within the framework of the authentification are indicated, a time stamp TS as well as a random number RN.

Further, a digital signature DS is provided as trailer information TI. For example, the asymmetrical RSA method can be employed as algorithm for the authentification.

Access Control for Management Information

The SNMPv1 access control is based on two mechanisms. First, an access control value is allocated to each managed object in an MIB, this comprising one of the three following values:

- read only,
- read-write,
- write only,
- not accessible.

Second, what is referred to as an MIB viewed together with the respective access rights is allocated to each community in the SNMPv1 agent configuration. An

MIB view contains a prescribable plurality of object identifiers that indicate the respective sub-trees or what are referred to as leaves of the SNMP registration tree.

The respective access rights comprise one of the following values:

- read only,
- write only,
- read-write,
- none.

Security of the Data Integrity of an SNMP Request

A mechanism for the cryptographic protection of the data integrity is utilized for securing the data integrity. Data integrity checksums are formed over the entire SNMPv1 request or over a part thereof for this purpose. This can ensue, for example, with the DES in what is referred to as the cipher block chaining mode (CBC mode). The employment of a 64 bit long initialization value is required for this specific mechanism, this having to be known to every party of the respective security group. The initialization value is part of the algorithm parameter AP that is employed in the header information HI of the cryptographically processed message KBN (see Figure 5). Further, the header information HI comprises a key identifier KID as well as an algorithm identifier AID whose functionality is the same as in the authentification.

Further, an integrity checksum ICV is provided in a trailer information TI.

Encryption of SNMPv1 Requests

Confidentiality of the transmitted SNMPv1 data can ensue in a way similar to the protection of the data integrity. For example, the DES method in the CBC mode can again be employed for the encryption. In this case, an initialization value is again required as algorithm parameter AP and a header information HI of the cryptographically processed message KBN is required (see Figure 6).

A key identifier KID as well as an algorithm identifier AID having the above-described functionality are again provided in the header information HI.

Further, mechanisms for logging the communication as well as for outputting an alarm when attempted attacks are found can be provided.

The method and the computer system can be very advantageously employed within the framework of a scenario wherein a vendor of a communication network makes bandwidth of the communication network available to a service provider who makes additional services available to third parties that do not provide the communication network in and of itself. In this context, the method as well as the computer system can advantageously serve, for example, for controlling or for accounting for the resources made available by the vendor of the overall communication network. In this case, the manager will be realized on a computer of the vendor of the overall communication network and an agent will be realized at the respective provider of additional services.

It is provided in one version of the above described exemplary embodiment to directly encode the reply message without waiting for a fetch message and to send it to the first computer unit. The following steps are thus not required in the second computer unit:

- the encoding of a fetch message according to the encoding format of the network protocol in the first computer unit, with which the cryptographically processed reply message is requested from the second computer unit;
- the transmission of the fetch message from the first computer unit to the second compute unit; as well
- the reception of the fetch message.

The analogous case applies to the computer system.

Clearly, the method can be described such that a cryptographic process is applied to the standard-conforming network protocol, for example the SNMPv1 protocol, being applied to the respective SNMP request or CMIP request as well, a cryptographic protection of the SNMP request or, respectively, the CMIP request being achieved with this. In order, however, to enable the employment of standard-conforming SNMP methods, the cryptographically processed message is again

encoded with the respective encoding format of the network protocol. This corresponds to a "double" application of the respective network protocol to the message to be encoded.